Sustainable Development through Holistic Multi-Paradigmatic Integrated Modeling and Decision Making

Full Paper

Minh Do
University of Auckland
Mdo581@aucklanduni.ac.nz

Valeria Sadovykh
PricewaterhouseCoopers and
University of Auckland
valeria.a.sadovykh@sg.pwc.com

David Sundaram
University of Auckland
d.sundaram@auckland.ac.nz

Abstract

Sustainability, sustainable development, and transformation are about balancing the economic, environmental and social aspects of organisations and their operations. Existing systems do not comprehensively support sustainable transformation nor do they allow decision makers to explore interrelationships and influences between the sustainability dimensions. This leads to silo based decision making where vision and strategies are not mapped to execution, and sustainability modelling processes are uncoordinated, lack holism, are biased and myopic. One of the critical challenges is the lack of processes and systems that allow the integration of models belonging to disparate paradigms in a holistic manner. To overcome these problems this research proposes a holistic, multi-paradigmatic, integrated modelling and decision making (HOMIMD) approach for sustainable development and transformation. We apply the HOMIMD processes, and requirements to a warehouse management problem and leverage optimization (AIMMS), systems dynamics (iThink), and data mining (SPSS Modeler) approaches, techniques, and systems to solve the problem.

Keywords

Sustainability, development, modelling, optimization, system dynamics, data mining, integrated, holistic, decision making, warehouse management

Challenges to Sustainable Development and Transformation

Sustainable development attempt to integrate, interweave and balance economic, environmental and social dimensions (Heemskerk et al. 2002), commonly termed as TBL or triple bottom line, into decision making (Elkington 2004). Sustainable management of businesses is the roadmap to achieving the present needs of society without compromising the opportunities of future generations (Hart and Milstein 2003). This is true at the micro level in the life of individuals and families and at the macro level in organisations, supply chains and societies as a whole.

The vision of our research is to design and implement processes and systems that will enable individuals, families, organisations, supply chains, and ultimately society to become more sustainable in terms of economic, environmental, and societal dimensions. The key challenges that we face are: (a) the presence of complexity (b) lack of holism (c) lack of integration (d) silo mentality in organisations (e) uni-dimensional reporting (f) biased myopic reporting (g) sustainability visions and strategies not aligned nor in sync with actions and (h) lack of a modeling driven systems thinking approach to sustainability, development, and transformation. In this paper we propose a holistic multi-paradigmatic integrated modeling and decision making (HOMIMD) approach to address these eight challenges. We illustrate the HOMIMD approach through its application in the context of warehouse management.

Key phases involved in the modeling life cycle (Hürlimann 2013) are the formulation and use of the model, analysis of the results of the model, reformulation of the model if necessary, storage and retrieval of the model, use of the model in conjunction with other models, use of the model along with other models as building blocks in the creation of larger, more complex models, and finally termination of the model. Most of the research effort in the area of MS/OR modeling in the past has concentrated on the discovery and implementation of efficient solutions to models with little attention given to the management of the modeling life cycle in an organisational context (Iyer et al. 2005). Most modeling software support only some phases of the modeling life cycle (Melão and Pidd 2000). These problems in turn lead to a variety of systems with different data format requirements, different syntax, etc. being linked together to achieve more comprehensive modeling life cycle support. These kinds of patchwork systems result in perennial interface problems as well as the need for the end user to know and keep up to date with the different syntax that each of the components uses.

Modeling in MS/OR has been in most cases a one-shot low-productivity exercise. Once built and used, the models were rarely used again since the conditions/premises on which the models were built had changed in time (Sundaram and Wolf 2009). Many modeling systems support the situation where the same model structure is instantiated with different data. But most systems do not allow the easy creation of versions of models where the structure of the model changes. Hence there is very little support for the evolution of models. In most systems there is reuse of the solver but very little reuse of the model or the data used to instantiate the model. This leads to redundancy, inconsistency, demand for different skills, and hence low productivity (Geoffrion 2013). This also leads to decision and policy makers finding it difficult to understand the models built, build models on their own, use the models, and modify existing models. Lack of standardised easy to use interfaces between models and data and between models and solvers lead to time-consuming tasks of specially preparing the data to suit the specific model and solver. This again requires specialised skills (Geoffrion 2013) preventing average decision makers from using the system.

Most modeling software are very domain specific (production/finance/marketing domain specific) and cannot be used outside the domain. Apart from this many modeling systems implement only one modeling paradigm (like Linear Programming or Simulation). Most modeling software lack model integration support (Argent 2004). They have some ability to consolidate results from different models, but most do not support the integration of models belonging to different model classes, or models belonging to different modeling paradigms, or deep integration where intermediate results of one model affect the selection or execution of another model. Most modeling systems also lack the capability to integrate simple model structures to form complex model structures.

There is a need for an HOMIMD platform where all the activities involved in modeling can be carried out within one seamless environment overcoming problems of interfacing data with models and solvers with models. In the next section we describe the HOMIMD approach followed by an implementation of the HOMIMD approach.

Holistic Multi-paradigmatic Integrated Modeling and Decision Making (HOMIMD) approach

The Overall HOMIMD Process and Requirements

The overall HOMIMD process involves a number of phases that provide cradle-to-grave modeling support (Geoffrion 2013; Hürlimann 2013). The phases are:

- Formulation of the model - support the modeler to create the model
- Integration of the model with data/instantiation of the model - support the modeler in the instantiation of the model
- Integration of the model with appropriate solvers - allow the modeler to link to a variety of solver technologies
- Use/execution of the model - allow the modeler to solve the model/problem using a particular instance of the model
- Analysing, reporting, and explaining the results of the model
- Perform *what-if* analysis on the model by changing the solvers and/or data and/or model versions

- Reformulation of the model if necessary - reformulation of the model by changing either its structure or behaviour or both
- Storage of the model - storage of the instance of the model as well as the structure and behaviour of the model
- Retrieval of the model - retrieval of model instance information as well as information relating to the structure and behaviour of the model
- Termination or removal of the model - removal of the model instance as well as removal of a particular behaviour of the model or part of the model structure or the whole model itself

These phases do not always occur sequentially and several phases would be revisited over the modeling life cycle. These phases are generic and apply to the solving of all kinds of problems and support decision making in various domains and using various paradigms. Hence these phases are ideal for modelling sustainability problems and decisions as they span multiple domains, paradigms, and philosophies. In the following paragraphs we explore the details and issues involved in the process of modeling.

Model formulation is an exercise through which the user specifies to the modeling system the structure of the model and the relationships between the components of the model. The system should allow the modeler complete flexibility to specify the components of the model and their interrelationships. Formulation of the model could be an automatic process or could be a process that is completely driven by the user/modeler. Between these two extremes there could exist a range of scenarios with varying levels of model formulation support from the system. The formulation phase could be further divided into problem elicitation/problem specification, model-type identification, and creation of the model. Some of the steps in model formulation could be automated (for example, based on the problem elicitation the modeling system could identify the type of model and formulate the model). There are advantages as well as disadvantages in such automation. Automation of such a phase implies either that the system has a huge knowledge base or that the system is narrowly focused. The advantage of such a system is that it is user-friendly and makes the process of modeling very much easier.

Another aspect of great importance is the ready and easy access to extensive corporate data (Dolk 2000). It is primarily to solve this problem that many modeling systems have tried database approaches, wherein the DBMS acts as the host for the modeling system. The provision of complete data management features as part of the modeling system is highly desirable (Geoffrion 2013). The process of instantiation of the model should be trivial if the above-mentioned feature is present.

Another issue closely related to model formulation as well as model instantiation is the checking of the integrity/validity of the model. This could mean checking the validity of the structure and/or behaviour of the model as well as checking the validity of the model instance.

The integration between models and solvers is quite important but difficult. The association of solvers to solve a particular model is not difficult. But the transfer of the model instance to the solver when a model is executed is complex, and involves the transformation of data stored in model-oriented format to a format that is specific to the solution algorithm. Once the solution is obtained, the result given by the algorithm has to be transformed into a format that is understandable to the model. This transformation process is not trivial and can be challenging. An important feature of a modeling system is the library of solvers that are made available (Geoffrion 2013). The modeler should have flexibility in the selection of solvers. The library should include solvers belonging to a variety of domains and different paradigms. The system should also possess solver interface routines that are general enough to link models with a variety of solvers.

There should be flexibility in the execution of the model. It should be possible for the model to be executed without updating the corporate data or even the model instance. The modeler should have the flexibility to update the results of the model execution into the model instance or to feed the results of the model execution as input to other models.

Support for the analysis of the results of the execution of the model is desirable in many cases. But this again, like the automatic model formulation features, has the same advantages and drawbacks. Apart from analysis of the results there should be a facility to conduct *what-if* analysis, goal-seeking, etc. There should also be provision for using the model in different contexts, which could even mean changing the direction of computation. The modeling system should also allow the modeler to explore various scenarios by varying the model parameters, changing the solvers used, and/or changing the model used.

Reformulation of the problem can be just a change of values in the model instance or a change in the structure of the model or a change in the behaviour of the model. To change the values in a model instance is easy and does not pose any problems. However, changing the model structure or behaviour poses problems such as versioning of models, management of model versions, keeping track of changes to the model, etc. There should be provision to not only store the different versions of the model but also to manage them. Object-oriented concepts such as inheritance and abstractions such as generalisation and aggregation could be used in versioning as well as in version management.

Model storage implies storage of the instance as well as storage of the model structure/schema. There should be a provision for the storage of model instances so that it is possible to compare model instances and retrieve instances for later use. Information relating to the schema/structure of the model should be stored in some form so that we can use the schema/structure in another model or as part of another model or modify it to create another version of the model. Storage of the model structure and instance allows sharing them among other decision makers/modelers within and outside the organisation. It is also desirable that the sharing be controlled, so that we can selectively give permission to users to view and/or use the models. Granting permissions for viewing and/or using different levels of the model by users with different privileges should be possible.

There should be provision for the removal/termination of model instances as well as the removal of the whole model including the model structure and model behaviour if necessary. Another issue that is of relevance is the documentation of the process of modeling. This not only means logging/keeping track of the decision maker's actions but also capturing the changes that occur to a model with varying parameters over a period of time. This helps us in gaining insight into the usage of models, data, parameters, and solvers in different contexts. It also provides data that would be valuable in understanding how the usage of the system changes the behaviour of the decision maker over time.

It is desirable that there is one common language that allows us to uniformly manage models, data, and solvers without bias towards a problem domain, modeling paradigm, or solver technology (Geoffrion 2013). The modeling language should be powerful enough to support the complete modeling life cycle from creation of the model to termination of the model. This language could be graphical (preferable) or textual.

Integration of Models

One of the key steps in the HOMIMD process is integration of multi-paradigmatic models in a holistic and flexible manner. Such integration could be *ad hoc* - where we link the outputs and inputs of two or more models in a temporary manner or *permanent* - where we use individual models or parts of individual models as components in the construction of new and larger models. The first type of integration is more like process integration whereas the latter type is schema integration. We discuss the ad hoc and permanent integration of models in the following sections.

Ad hoc Integration

Ad hoc (or temporary or *on the fly*) integration is an essential part of modeling and the system should facilitate such integration. In this type of integration the inputs to models could be from other models or from corporate data sources or directly from the user. The results/outputs of the model could be used to update corporate data sources or could serve as inputs to other models. There can be ad hoc integration at various levels of complexity, and four levels of integration complexity can be identified:

1. Integration of model instances is one of the simplest types of integration (Geoffrion 2013). This integration can be consolidation or aggregation of the values obtained by the execution of different model instances.
2. Integration of models belonging to different model classes but within the same modeling paradigm (Geoffrion 2013). An example of this is the combining of two or more LP models into a comprehensive or meta-model.
3. Integration of models belonging to different modeling paradigms (Geoffrion 2013). For example, integration of an LP model with a non-linear programming (NLP) model or a Forecasting model with a Simulation model.

4. The models and their integration discussed above are either of the nature of 'consolidation' where results from multiple models are consolidated or of 'pipelining' where results from one model are fed to another model. Makowski (2005) identifies another type of integration called 'splicing' where the final results or the intermediate results of a model affect the selection of other models. That is, there is an element of dynamism in the integration of models. Models may also be so closely linked that the partial results from one model may feed to another model and vice-versa.

Within each of these levels of integration complexity, there could be further sub-levels. Geoffrion (2013) suggests ten such levels. Most attempts in IMMS at some kind of integration have been at the first and second levels mentioned above. But the third and definitely the fourth levels mentioned above have received little attention in literature. Makowski (2005) outlines a model manipulation language based on the idea of communicating sequential processes. They identify two critical aspects that should be defined when specifying how individual model components are to be integrated to form a meta-model: (1) The input/output relationship between model component variables specifying which outputs from one model are going to serve as inputs to other model components. (2) The order in which the model components have to be executed and the timing of the dynamic interaction of model components.

Makowski (2005) proposes a process-oriented conceptualization where model components are considered as processes that communicate by sending and receiving messages. They propose constructs that allow model components to be used as building blocks in ways unforeseen when the components were originally developed. Current approaches do not consider issues of dynamic variable correspondence between models and synchronisation in the execution of multiple models.

From the software engineering field there have been quite a few efforts towards the reuse of software components. These ideas could be of use in integration of models. Coulange (2012) suggests the use of a *library interconnection language* (LIL) to assemble large programs from existing entities. Llorens et al. (2006) suggest the use of *module interconnection languages* (MIL) to provide formal grammar constructs for deciding the various module interconnection specifications required to assemble a complete software system. The ideas and concepts used in these systems could definitely prove beneficial in the development of similar interconnection languages for the integration of simple models to form more complex models belonging to multiple paradigms.

Permanent Integration

Permanent integration involves the creation of a new model by integrating the structures and behaviours of two or more models or the components from two or more models. Key issues in permanent integration become apparent when we look at various proposals for permanent integration of models. Geoffrion (2013) proposes a five-step approach for integrating structured model schemas, which is a labour intensive, non-trivial exercise. The steps are 1) make appropriate structural changes to achieve consistency between individual models and also to suit the new roles they will play in the integrated model 2) resolve naming and dependency conflicts and identify and merge related model structures 3) join the two models 4) revise integrated model so that it satisfies structured modeling criteria and finally 5) make cosmetic changes.

The IS discipline has also studied the problem of schema integration in the context of databases (Schmitt and Saake 2005), and quite a few of the problems and issues that are raised, are very similar and relevant to model schema integration. There are three steps in the process of database schema integration 1) comparison of schemas and resolution of naming (homonyms and synonyms) and structural conflicts (type, dependency, key and behaviour) 2) conforming or aligning the schemas by resolving the conflicts and finally 3) merging and restructuring of the schemas in such a way that it is complete, minimal, and understandable. Lessons learnt by the database community in the integration of database schemas could be useful in the integration of model schemas.

In the next section, we will illustrate how the HOMIMD approach is applied to a sustainable warehouse management context using the ad hoc integration proposed in the HOMIMD method. However, it is still able to produce robust results to address economic, environmental and social issues in decision making. Future research can be conducted to examine how a higher form of integration, for example permanent integration, can aid organizations in solving more complex problems.

Implementation of the HOMIMD Approach

Due to a rapidly changing business environment, companies nowadays often confront problems that are too complex for human minds to comprehend and make effective decisions in a timely manner (Vercellis 2011). As a result, many companies, especially in developing countries, focus only on the economic dimension of the decision making processes which is to minimize cost or maximize profit (Baughn and McIntosh 2007). Sooner or later, it will have negative effects on future generations.

To support key personnel in companies in making effective and timely decisions in a sustainable manner, the HOMIMD approach can be used to utilize the strengths of multiple modeling paradigms (eg. data mining, mathematical modeling and system dynamics) in an integrated manner to provide decision makers with a holistic view of the problems and their inter-relationships.

In this example, we consider a theoretical company that has warehouses in two locations (Auckland and Wellington). Their customers are located in Hamilton, Rotorua, Whangarei, Palmerston North, and Tauranga. The company wishes to find the optimal amount of products that need to be transported from each warehouse to customers' locations to maximize their profit taking into account the social and environmental performance of their decisions.

The social aspect of the case is modeled by SPSS Modeler which is a data mining tool to predict which trucks have the highest probability of mechanical failures happening on the road so that proactive maintenance can be carried out to reduce traffic accidents and fatalities caused to civilians. Data mining is particularly useful in analyzing, identifying patterns from a vast amount of historical data companies gathered and then converting them into useful information to support decision making processes (Han et al. 2011). Besides, data mining with predictive analysis is an excellent tool for predicting the impact of decisions on society and environment.

The output of this model serves as an input value and a constraint for the AIMMS (mathematical) model with the purpose of maximizing the company's profit subject to supply, demand constraints and the number of trucks available to transport from the warehouse's location to customers'. Mathematical modeling, on the other hand, is of great importance in assisting companies in finding optimal solutions for a problem subject to a set of constraints. It is widely used for optimizing companies' operational activities (Scarf 1997) to minimize cost or to maximize profit.

Similarly, the output of the AIMMS model which is the amount of profit obtained and the number of trucks required to fulfill customers' orders serves as the input figure for the iThink model (system dynamics). System dynamics methodology is chosen as it provides decision makers with a holistic view of a problem that takes into account feedback loops of actions and inter-connectedness of multiple parts in a system (Maani and Cavana 2007). In some cases, the interactions between the parts are more important than the problem itself. Hence, system dynamics is a widely used tool to address dynamically complex environmental problems (Winz et al. 2009). In this example, the iThink model is used to identify the number of seedlings the company needs to plant each month to absorb the amount of carbon dioxide emitted from their fleet of trucks. Figure 1 depicts how the HOMIMD approach is applied to support sustainable decision making. All three modeling paradigms mentioned above are integrated to aid companies in solving the transportation problem without damaging society nor the environment.

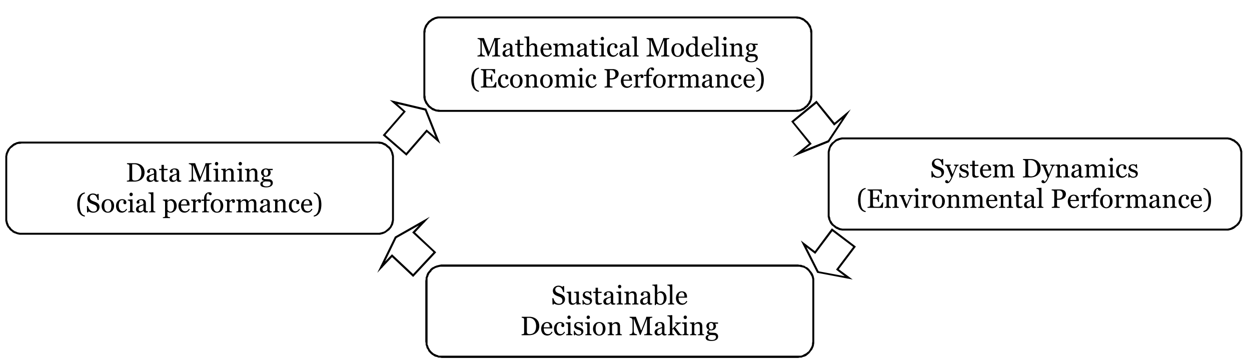

Figure 1: HOMIMD approach to support Sustainable Decision Making

Social Dimension - Data Mining Paradigm - SPSS Modeler

Decision tree is a method that consists of decision rules to split data into groups. The objective of the decision tree is to discover a set of rules that can be used to predict outcomes from a set of input variables (Pradhan 2013). Decision tree has been widely applied in several practical circumstances for prediction and classification (Murthy 1998). In this example, a decision tree is built based on C5.0 algorithm which is well known for its accurate and efficient results (Pandya and Pandya 2015). Half of the historical data is used to build the model; the other half is used to test the validity of the model as illustrated in Figure 2. The model is then used to predict the probability of mechanical failure happening to the fleet of trucks (Figure 3). The output of this model serves as an input value and a constraint for the following AIMMS model.

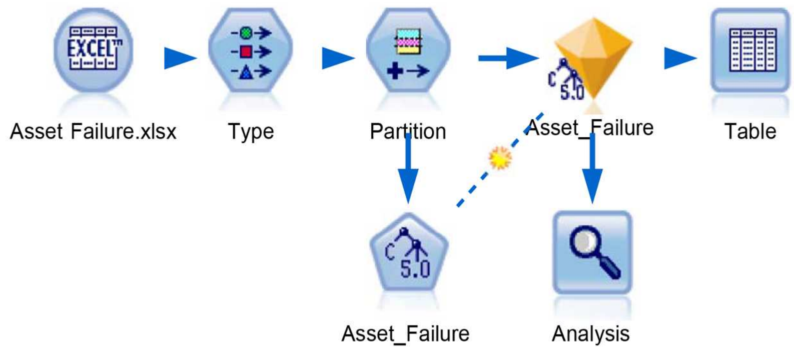

Figure 2: Building and training the SPSS Modeler Data Mining model

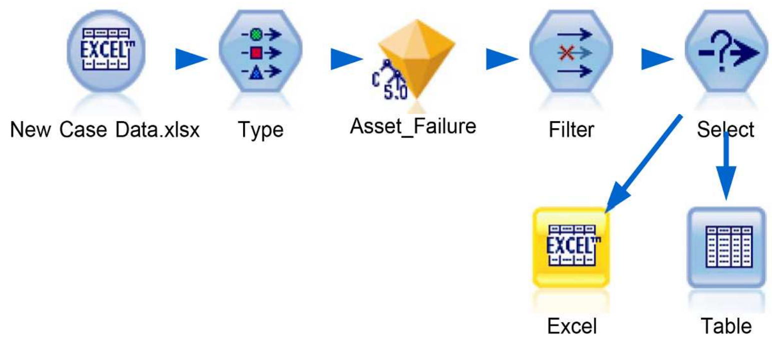

Figure 3: Predicting the probability failure of trucks and impact on the Social Dimension

Economic Dimension – Mathematical Modeling Paradigm - AIMMS

The AIMMS model (Figure 4) is designed to help managers find optimal solutions for their transportation problem to maximize profit subject to supply and demand constraints (Chanas and Kuchta 1996). In this model, we expand the transportation problem by adding one more constraint which is the number of trucks available to transport from the warehouse's location to customers'.

The output of the model that is the amount of profit obtained and the number of trucks required to fulfill customers' orders serves as the input for the iThink model. The mathematical model is depicted as follows:

Max: $\sum_{ab} P_{ab} \,.\, X_{ab}$

Subject to: $\sum_{b} X_{ab} \le S_a;\ \sum_{a} X_{ab} \le D_{b;}\ T_u \le T_a$

With: X_{ab}: The amount of cargo shipped from warehouse A to customer B

P_{ab}: The unit profit when products are shipped from warehouse A to customer B

S_i: The total amount of products can be supplied at warehouse a

D_b: The total amount of products required by customer B

T_u: The number of trucks is used to transport products from warehouse A to customer B. It is assumed that each truck can carry Y unit of products per trip and the total km each truck travel should not exceed Z (km).

T_a: The number of trucks available in the company

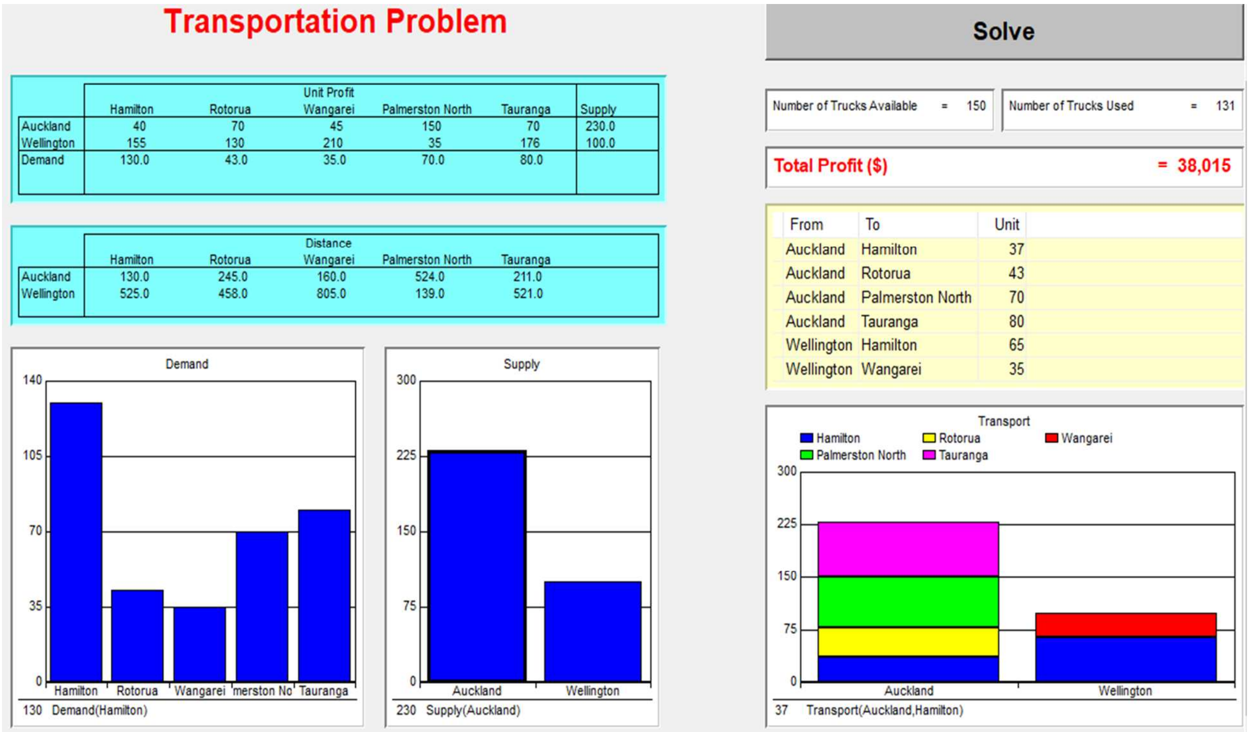

Figure 4: AIMMS Optimization model to evaluate the Economic Dimension

Environmental Dimension – System Dynamics Paradigm - iThink

Based on the number of trucks required to transport products from the two warehouses to customers' locations, the iThink model (Figure 5) is built to calculate the number of trees that need to be planted to absorb carbon dioxide emitted from the fleet of trucks. It is assumed that the company has a plan to plant a certain amount of seeds each month. Over months, those seeds will grow up to become saplings and then mature trees to absorb the carbon dioxide. An interface of the model (Figure 6) has also been designed to assist users in changing input data of the model, for example, the number of seeds plants each month, carbon dioxide emitted per truck or the number of trucks used. By changing the input data, the company can measure their environmental performance to support decision making process.

This example shows how the HOMIMD approach can be used to assist managers in making sustainable decisions in an effective and timely manner by utilizing the power of multiple modeling paradigms in an integrated manner. The output of one model serves as the input for another model and so on. All three models are closely connected to help decision makers find optimal or satisficing solutions for the simple transportation problem without harming society nor the environment.

Conclusion

Sustainable development is about keeping in mind and balancing the economic, environmental, and social dimensions of individuals, families, organizations, supply chains, communities, nations, and indeed society as a whole. Current approaches, models, processes, and systems tend to be myopic, monolithic, and limited and explore one and at the most two of the sustainability dimensions. There are very few systems that explore the interrelationships of all three dimensions in an integrated, interwoven and holistic manner. This obviously leads to sub-optimal silo based uncoordinated decisions and actions that are in no way reflective of the sustainability visions and missions espoused by organizations or nations. One of the key challenges is the lack of systems to allow the representation, execution, and integration of models belonging to disparate paradigms in a holistic manner.

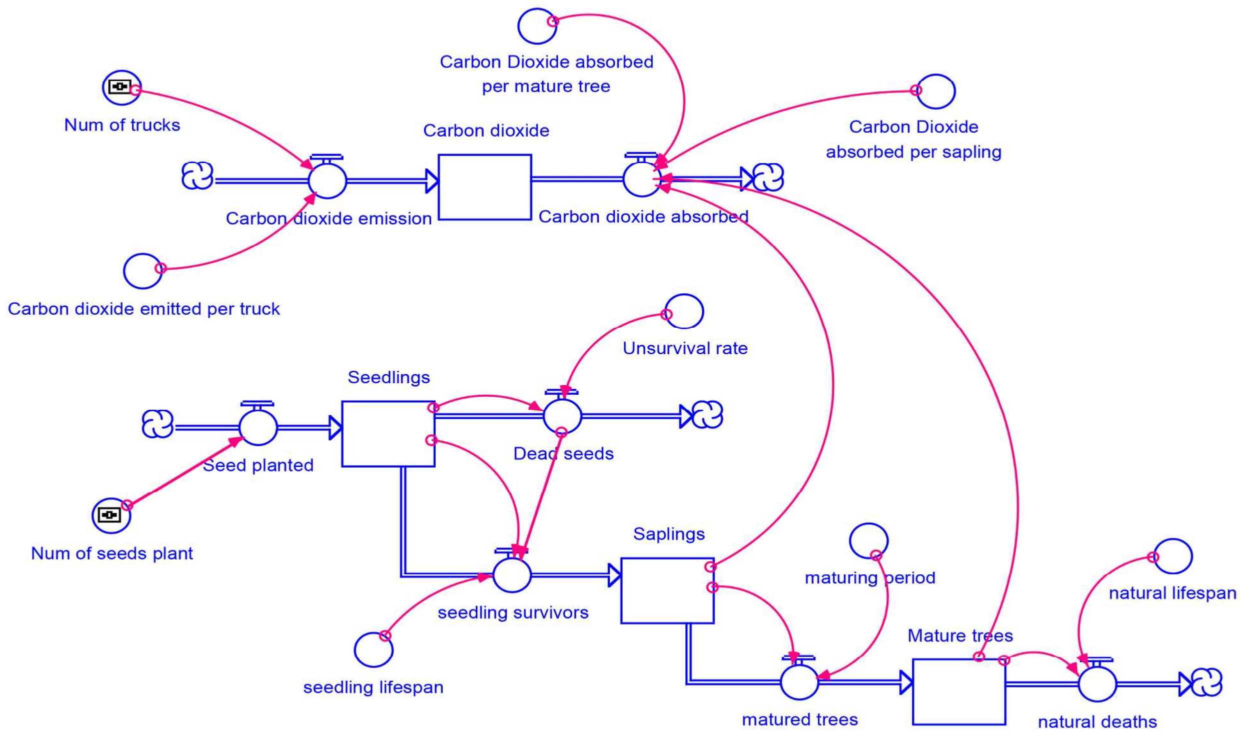

Figure 5: iThink Simulation model to evaluate the Environmental Dimension

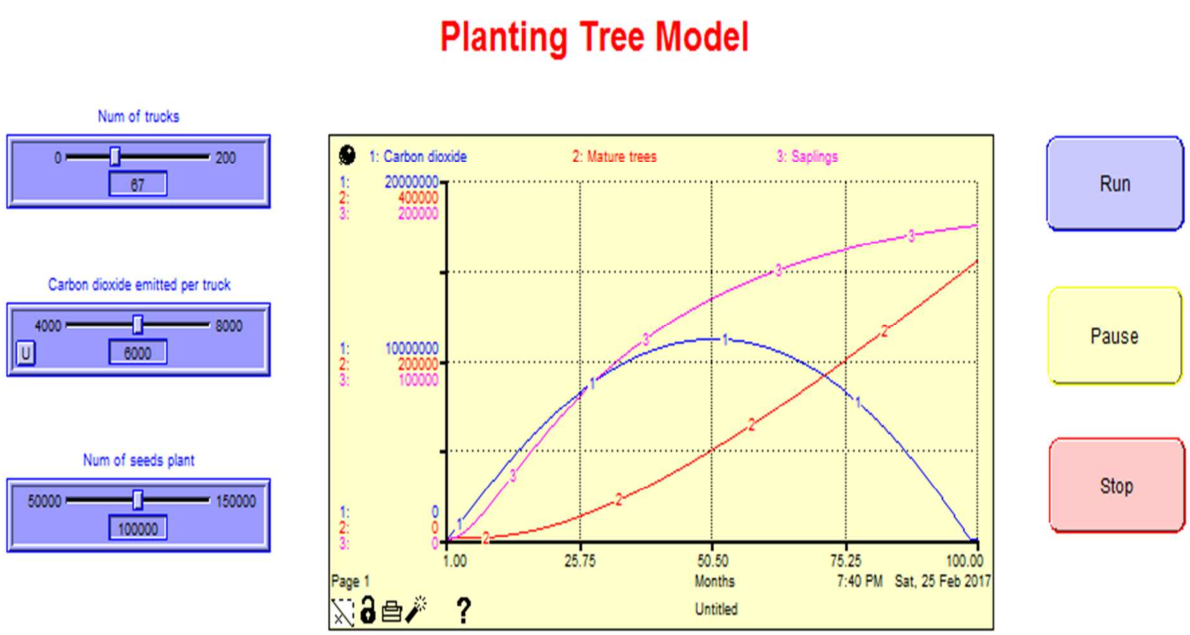

Figure 6: What-If Analysis using iThink model

To address these challenges we propose a holistic, multi-paradigmatic, integrated modeling and decision making (HOMIMD) approach that could be used for the sustainable development and transformation individuals, organisations, and indeed society as a whole. We applied the HOMIMD principles, processes, and requirements to a warehouse management problem. Furthermore we leverages optimization (AIMMS), systems dynamics (iThink), and data mining (SPSS Modeler) approaches, techniques, and systems in an interwoven, integrated, and holistic manner to solve the problem.

REFERENCES

Argent, R. M. 2004, "An Overview of Model Integration for Environmental Applications—components, Frameworks and Semantics," Environmental Modelling & Software (19:3), pp. 219-234.

Baughn, C. C., and J. C. McIntosh. 2007, "Corporate Social and Environmental Responsibility in Asian Countries and Other Geographical Regions," Corporate Social Responsibility and Environmental Management (14:4), pp. 189-205.

Chanas, S., and D. Kuchta. 1996, "A Concept of the Optimal Solution of the Transportation Problem with Fuzzy Cost Coefficients," Fuzzy Sets and Systems (82:3), pp. 299-305.

Coulange, B. 2012. Software Reuse, Springer Science & Business Media.

Dolk, D. R. 2000, "Integrated Model Management in the Data Warehouse Era," European Journal of Operational Research (122:2), pp. 199-218.

Elkington, J. 2004, "Enter the Triple Bottom Line," The Triple Bottom Line: Does it all Add Up (11:12), pp. 1-16.

Geoffrion, A. M. 2013. "Structured Modeling," Encyclopedia of Operations Research and Management Science, Anonymous Springer, pp. 1499-1503.

Han, J., J. Pei, and M. Kamber. 2011. Data Mining: Concepts and Techniques, Elsevier.

Hart, S. L., and M. B. Milstein. 2003, "Creating Sustainable Value," The Academy of Management Executive (17:2), pp. 56-67.

Heemskerk, B., P. Pistorio, and M. Scicluna. 2002. Sustainable Development Reporting: Striking the Balance, World Business Council for Sustainable Development.

Hürlimann, T. 2013. Mathematical Modeling and Optimization: An Essay for the Design of Computer-Based Modeling Tools, Springer Science & Business Media.

Iyer, B., G. Shankaranarayanan, and M. L. Lenard. 2005, "Model Management Decision Environment: A Web Service Prototype for Spreadsheet Models," Decision Support Systems (40:2), pp. 283-304.

Llorens, J., J. M. Fuentes, R. Prieto-Diaz, and H. Astudillo. 2006. "Incremental Software Reuse," pp. 386-389.

Maani, K., and R. Y. Cavana. 2007. Systems Thinking, System Dynamics: Managing Change and Complexity, Prentice Hall.

Makowski, M. 2005, "A Structured Modeling Technology," European Journal of Operational Research (166:3), pp. 615-648.

Melão, N., and M. Pidd. 2000, "A Conceptual Framework for Understanding Business Processes and Business Process Modelling," Information Systems Journal (10:2), pp. 105-129.

Murthy, S. K. 1998, "Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey," Data Mining and Knowledge Discovery (2:4), pp. 345-389.

Pandya, R., and J. Pandya. 2015, "C5. 0 Algorithm to Improved Decision Tree with Feature Selection and Reduced Error Pruning," International Journal of Computer Applications (117:16).

Pradhan, B. 2013, "A Comparative Study on the Predictive Ability of the Decision Tree, Support Vector Machine and Neuro-Fuzzy Models in Landslide Susceptibility Mapping using GIS," Computers & Geosciences (51:pp. 350-365.

Scarf, P. A. 1997, "On the Application of Mathematical Models in Maintenance," European Journal of Operational Research (99:3), pp. 493-506.

Schmitt, I., and G. Saake. 2005, "A Comprehensive Database Schema Integration Method Based on the Theory of Formal Concepts," Acta Informatica (41:7), pp. 475-524.

Sundaram, D., and E. Wolf. 2009. "Enterprise Model Management Systems," pp. 5.

Vercellis, C. 2011. Business Intelligence: Data Mining and Optimization for Decision Making, John Wiley & Sons.

Winz, I., G. Brierley, and S. Trowsdale. 2009, "The use of System Dynamics Simulation in Water Resources Management," Water Resources Management (23:7), pp. 1301-1323.